7*u*

UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/040,276	11/09/2001	Daniel S. Gritter	POU901034US1	9648

46369 7590 01/19/2007		EXAMINER
HESLIN ROTHENBERG FARLEY & MESITI P.C.		YIGDALL, MICHAEL J
5 COLUMBIA CIRCLE		
ALBANY, NY 12203	ART UNIT	PAPER NUMBER
	2192	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/19/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

PTOL-90A (Rev. 10/06)

	Application No.	Applicant(s)
	10/040,276	GRITTER, DANIEL S.
Office Action Summary	Examiner	Art Unit	
	Michael J. Yigdall	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *12 October 2006*.

2a) ☒ This action is **FINAL**.   2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *45-64* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) *45-64* is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☒ The drawing(s) filed on *12 October 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1.   This Office action is responsive to Applicant's submission filed on October 12, 2006.

Claims 45-64 are pending.

Response to Arguments

2.   Applicant's arguments have been fully considered but they are not persuasive.

Applicant contends that the claims comply with 35 U.S.C. 112, first paragraph, because

the limitation "said program being absent embedded debug commands" is supported by and

reflected in the specification as originally filed (remarks, page 10, last paragraph). The basis of

Applicant's argument is that the specification as originally filed made no reference to and did not

include embedded debug commands (remarks, page 10, first paragraph).

However, the examiner does not agree with Applicant's conclusion. It is because the

specification as originally filed made no reference to "embedded debug commands" that it does

not provide the necessary support for this limitation.

37 CFR 1.75(d)(1) states, "The claim or claims must conform to the invention as set forth

in the remainder of the specification and the terms and phrases used in the claims must find clear

support or antecedent basis in the description so that the meaning of the terms in the claims may

be ascertainable by reference to the description."

Here, the term "embedded debug commands" does not have clear support or antecedent

basis in the description, and thus the meaning of the term "embedded debug commands" is not

ascertainable by reference to the description. See also MPEP § 608.01(o). In fact, it is the

Shridhar reference (U.S. Patent No. 5,815,714) that provides antecedent basis for the term

"embedded debug commands," rather than Applicant's specification as originally filed.

Applicant's response filed on December 14, 2004 indicates that the limitation "said program

being absent embedded debug commands" was added to the claims so as to distinguish the

claimed subject matter over Shridhar.

Moreover, Applicant's reasoning is that the specification depicts examples of source code

with no debug commands, and thus provides support that the program does not include

embedded debug commands (remarks, page 10, third paragraph). However, there are many

things that are not listed in the depicted examples of source code. Applicant's reasoning would

also suggest that the specification somehow provides support for the limitation "said program

being absent a kitchen sink," but this is clearly inconceivable.

Applicant notes that Olsen teaches that a breakpoint instruction is substituted for the

machine code instruction positioned at the HW and the machine code instruction at that position

is stored in a table for later use (remarks, page 12, last paragraph), but contends that the attributes

of the machine code instruction are not considered or used in any manner to form a profile or for

subsequent comparisons (remarks, page 13, top). Applicant further contends that this teaching in

Olsen is directed to insertion of a breakpoint instruction into binary, not source code, is not

automatic and does not use any characteristics of the replaced machine code instruction to

determine the selected step (remarks, page 13, first paragraph).

However, it seems that Applicant's arguments improperly attempt to narrow the scope of

the claims. Applicant states that the table in Olsen that is relied on as a basis for asserting that

Olsen creates an instruction profile is nothing more than a means of associating the inserted

breakpoint with the entire machine code instruction it replaced rather than any of the attributes of

that instruction (remarks, page 13, first paragraph). The claims, however, recite merely that the "instruction profile" comprises "one or more attributes of one or more machine instructions generated for the selected step and one or more attributes of zero or more other machine instructions generated for the first version of source code" (claim 45). Thus, an instruction profile that comprises one attribute of one machine instruction generated for the selected step anticipates the recited instruction profile. The independent claims do not specify what does and what does not constitute an "attribute." Even if the table that Olsen creates includes an entire machine code instruction, inherently it still includes "attributes" of that machine code instruction. For example, Olsen illustrates an assembly code instruction "mov r4 2" (column 4, Table 1). The instruction comprises an operation "mov," a target register "r4" and an operand "2," each of which is reasonably considered an attribute or a characteristic of the instruction. Even in the form of binary machine code, the instruction still comprises those elements.

Furthermore, the claims do not positively recite that the breakpoint, once automatically restored, is inserted into source code. Rather, the claim recites "having a breakpoint that is set to a selected step of a first version of source code of a program" and "automatically restoring the breakpoint to the selected step of a modified program" (claim 45). Olsen teaches the same. Olsen teaches having a breakpoint that is set to a selected step P in the source code of a program (see, for example, column 12, lines 1-3). The program is subsequently modified such that the selected step is at different location within the modified program (see, for example, column 2, lines 63-67). Olsen teaches automatically restoring the breakpoint (see, for example, column 13, lines 8-12) to that location (see, for example, column 5, lines 28-30). Indeed, the insertion of the

breakpoint into the modified program (i.e., the restoration of the breakpoint) is done

automatically. Applicant does not provide any evidence to the contrary.

Applicant notes that the Office action stated that Olsen further teaches that restoring the

breakpoint comprises comparing one or more attributes of one or more machine instructions

generated for the source code with one or more attributes of an instruction profile created based

on the source code to determine the different location of the selected step, but again contends

that Olsen fails to define or teach the creation and use of an instruction profile as claimed

(remarks, page 13, third paragraph). Specifically, Applicant contends that in Olsen, the attributes

of the machine instructions captured in the block or blocks of interest are not considered or saved

and that no profile comparable to that of Applicant is therefore created or referenced in

attempting to restore a breakpoint after optimization (remarks, page 14, first paragraph).

However, the teachings of Olsen include several examples of referencing an "instruction

profile" to compare attributes of the machine instructions with attributes saved therein (see, for

example, column 12, lines 15-63). For instance, Olsen discloses, "Thereafter, the debugger

procedure 30 retrieves the commitpoint C for source position P from a table within the compiled

application 24 (step 102)" (column 12, lines 18-20). Olsen also discloses, "Source statement Si

and the corresponding machine code instruction i are retrieved from a table established by

compiler 26 during the compile action" (column 12, lines 37-39).

As alluded to above, it seems that Applicant's arguments improperly attempt to narrow

the scope of "instruction profile" as recited in the claims. The tables and other related

information that Olsen generates (see, for example, column 8, line 30 to column 9, line 14) are

reasonably considered as such an instruction profile. Although the claims are interpreted in light

of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In response to Applicant's characterization of Baker (remarks, page 15, last paragraph to page 16, last paragraph), the examiner first notes that other than the limitation of "the modified program having a second version of source code" (claim 45), it is Olsen that teaches or suggests the claimed invention.

Nonetheless, Baker does teach a first version of source code that is modified to provide a modified program having a second version of source code (see, for example, column 1, line 64 to column 2, line 8), and does teach comparing one or more attributes of one or more binary code instructions generated for the different versions of the source code to find similarities in the program, even without access to or knowledge of the source code itself (see, for example, column 3, lines 17-31). It is, in fact, one or more "attributes" of the binary code instructions that are compared because, as Baker discloses, the "binary files are disassembled and preprocessed using encoding to prepare the disassembled binary files for use in conjunction with similarity detection tools" (column 3, lines 23-26).

Applicant contends that Olsen and Baker are each trying to solve significantly different problems than Applicant and are not reasonably pertinent to the particular problem Applicant's invention solves (remarks, page 17, second paragraph).

However, as set forth in the Office action mailed on July 12, 2006, the examiner disagrees. Again, Applicant's invention "relates, in general, to the debugging of computer programs, and, in particular, to the restoring of debugging breakpoints, subsequent to modifying

code of a program" (specification, page 1, paragraph [0001]). Olsen is certainly pertinent to this problem. Olsen relates, in particular, to the restoring of debugging breakpoints, subsequent to an optimization modifying code of a program (see, for example, the abstract). Part of Applicant's invention pertains to comparing lines of code and finding matches in the code (drawings, pages 5-6, figures 5A and 5B). Baker is certainly pertinent to this problem. Baker pertains to finding similarities in programs (see, for example, the abstract).

In response to Applicant's other arguments against the *prima facie* case of obviousness set forth in the Office action (remarks, page 17, last paragraph to page 22, first paragraph), the examiner first notes that the test for obviousness is not that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981). Accordingly, in this case, the test for obviousness is what the combined teachings of Olsen and Baker would have suggested to those of ordinary skill in the art. Moreover, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

Olsen teaches "a method for debugging a machine code of a program that has been subjected to an optimizing action, wherein the machine code may have been reordered, duplicated, eliminated or transformed so as not to correspond with the program's source code

order" (column 2, line 63-67). Likewise, Baker teaches that "a single primary change to a source file, e.g. the insertion of a line of code, may result in many secondary changes to the compiled binary executable file as compared to the compiled binary executable file for the original, i.e., unaltered, source file" (column 1, line 65 to column 2, line 2).

A teaching, suggestion or motivation to apply the method of Olsen, found within the reference itself, is to restore a breakpoint that is set to a selected step in the source code of a program to the same step in the machine code of the program (see, for example, column 3, lines 2-9), wherein, as above, "the machine code may have been reordered, duplicated, eliminated or transformed so as not to correspond with the program's source code order." In terms of Baker, such changes are examples of the "many secondary changes" that may result from "a single primary change" to the program's source code. One of ordinary skill in the art would have appreciated that just as "an optimizing action" provides a modified program in Olsen, so too does "the insertion of a line of code" provide a modified program in Baker. It would have been obvious to one of ordinary skill in the art to apply the method of Olsen regardless of how or why the modified program is provided. Thus, as set forth in the Office action, the combined teachings of Olsen and Baker would have suggested the claimed subject matter to those of ordinary skill in the art.

Claim Rejections - 35 USC § 112

3.   The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4.   Claim 52 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the

written description requirement. The claim contains subject matter that was not described in the

specification in such a way as to reasonably convey to one skilled in the relevant art that the

inventor(s), at the time the application was filed, had possession of the claimed invention.

     Specifically, the claim recites the limitation, "said program is absent embedded debug

commands." Applicant's specification as originally filed does not include any description of

"embedded debug commands," and accordingly cannot provide the necessary support for this

limitation. The specification as originally filed fails to provide proper antecedent basis for the

term "embedded debug commands." See 37 CFR 1.75(d)(1) and MPEP § 608.01(o).

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

6.   Claims 45-64 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent

No. 6,263,489 to Olsen et al. (art of record, "Olsen") in view of U.S. Patent No. 6,282,698 to

Baker et al. (art of record, "Baker").

     With respect to claim 45 (currently amended), Olsen discloses a method of restoring

debugging breakpoints (see, for example, the abstract), said method comprising:

(a) having a breakpoint that is set to a selected step of a first version of source code of a

program (see, for example, column 12, lines 1-3, which shows having a breakpoint that is set to a

selected step in the source code of a program);

(b) creating an instruction profile for the selected step, said instruction profile comprising

one or more attributes of one or more machine instructions generated for the selected step and

one or more attributes of zero or more other machine instructions generated for the first version

of source code (see, for example, column 12, lines 3-15, which shows calculating an instruction

profile for the selected step that comprises one or more attributes of one or more machine code

instructions generated for the selected step and other instructions in the source code); and

(c) automatically restoring the breakpoint to the selected step of a modified program (see,

for example, column 13, lines 8-12, which shows automatically restoring the breakpoint, and

column 5, lines 28-30, which shows that the breakpoint is restored to the selected step), in

response to modification of the first version of source code to provide the modified program,

wherein the selected step is at a different location within the modified program (see, for example,

column 2, lines 63-67, which shows that the source code was optimized to provide a modified

program, and that the selected step does not correspond to the same location).

Olsen discloses that the automatically restoring comprises comparing one or more

attributes of the one or more machine code instructions with the one or more attributes of the

instruction profile created based on the source code (see, for example, column 12, lines 15-63) to

determine the different location (see, for example, column 12, line 64 to column 13, line 4), but

does not expressly disclose that the modified program has a second version of source code, and

that the automatically restoring comprises comparing one or more attributes of one or more

machine instructions generated for the second version of source code with one or more attributes of the instruction profile created based on the first version of source code.

However, Baker discloses comparing one or more attributes of the one or more binary code instructions generated for different versions of source code to find similarities in the programs, even if the source code is not accessible (see, for example, column 3, lines 17-31). In Baker, the first version of source code is modified to provide a modified program having a second version of source code (see, for example, column 1, line 64 to column 2, line 8). Baker further discloses disassembling the binary code instructions and preprocessing the disassembled instructions to create an instruction profile that comprises one or more attributes of the binary code instructions (see, for example, blocks 120 and 130 in FIG. 1).

One of ordinary skill in the art would have been motivated to apply Olsen's teachings regardless of how or why the program is modified. In other words, one of ordinary skill in the art would have been motivated to automatically restore a breakpoint to a selected step of a modified program, regardless of whether it is an optimization (see, for example, Olsen, column 2, lines 63-67), a modification of the source code (see, for example, Baker, column 1, line 64 to column 2, line 8), or some other operation that provides the modified program.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate Baker's instruction profile into Olsen's instruction profile, and to not only compare the one or more attributes of the one or more machine code instructions with the one or more attributes of the instruction profile created based on the first version of source code, as Olsen teaches, but to also do so when the machine code instructions are generated for a second version of source code, as Baker suggests. Baker's instruction profile

enables such a comparison even when the different versions of source code are not accessible, thus providing more opportunities to apply Olsen's teachings.

With respect to claim 46 (previously presented), the rejection of claim 45 is incorporated, and Olsen in view of Baker further discloses that the comparing comprises comparing one or more operation codes of the one or more machine instructions generated for the second version of source code with one or more operation codes of the instruction profile to determine which machine instruction of the modified program corresponds most closely to the selected step (see, for example, Baker, column 7, lines 11-30, which shows that the opcodes of the assembly instructions are included in the instruction profile and are compared).

With respect to claim 47 (previously presented), the rejection of claim 45 is incorporated, and Olsen in view of Baker further discloses that the instruction profile further comprises a source line number for the selected step and a length of the first version of source code (see, for example, Olsen, column 8, line 30 to column 9, line 14, which shows that the instruction profile further comprises compiler information that includes source line numbers and lengths of the source code), and wherein the automatically restoring comprises using the source line number and length to determine a starting point within the modified program to select the one or more machine instructions generated for the second version to be used in the comparing (see, for example, Olsen, column 12, lines 15-27, which shows that a starting point for the comparing is determined from the compiler information).

With respect to claim 48 (previously presented), the rejection of claim 45 is incorporated, and Olsen in view of Baker further discloses that the comparing yields one or more difference

counts and a difference count having a smallest value indicates the different location (see, for example, Olsen, column 10, lines 12-16, which shows that the comparing comprises finding the difference in counter values, and column 9, lines 51-54, which shows that the earliest or smallest difference indicates the different location).

With respect to claim 49 (previously presented), the rejection of claim 45 is incorporated, and Olsen in view of Baker further discloses that the different location is identified by a substantial match between one or more attributes of the instruction profile and one or more attributes of one or more machine instructions of the modified program (see, for example, Olsen, column 10, lines 53-63, which shows that a substantial match between the one or more attributes of the instruction profile and the one or more attributes of the machine code instructions of the modified program identifies the different location).

With respect to claim 50 (previously presented), the rejection of claim 45 is incorporated, and Olsen in view of Baker further discloses that the creating comprises choosing a number of machine instructions to be included in the instruction profile (see, for example, Olsen, column 12, lines 3-14, which shows choosing a number of machine code instructions to include).

With respect to claim 51 (previously presented), the rejection of claim 50 is incorporated, and Olsen in view of Baker further discloses that the choosing comprises:

(a) selecting a number of instructions to be included in a calibration profile (see, for example, Olsen, column 5, lines 8-23, which shows selecting a number of instructions to include in a calibration profile);

(b) generating the calibration profile for a chosen line of the program, said calibration profile having the selected number of instructions for said chosen line (see, for example, Olsen, column 4, line 66 to column 5, line 8, which shows calculating or generating the calibration profile for a chosen location);

(c) comparing one or more attributes of said calibration profile to one or more attributes of at least one line of code of the program to obtain a result (see, for example, Olsen, column 12, lines 15-63, which shows comparing one or more attributes of the calibration profile to one or more attributes of at least one line of code);

(d) determining whether the result is an unambiguous result (see, for example, Olsen, column 5, lines 38-45, which shows an ambiguous result, and column 5, lines 46-51, which shows determining whether the result is unambiguous); and

(e) repeating, zero or more times, said selecting, said generating, said comparing, and said determining until the determining indicates an unambiguous result, wherein the selected number of instructions increases at each iteration (see, for example, Olsen, column 9, lines 19-49, which shows repeating the steps zero or more times until the result is unambiguous), and wherein the selected number of instructions indicates, when there is an indication of an unambiguous result, the number of machine instructions to be included in the instruction profile (see, for example, Olsen, column 10, lines 4-11, which shows indicating the instructions to include in the instruction profile when the result is unambiguous).

With respect to claim 52 (currently amended), the rejection of claim 45 is incorporated, and Olsen in view of Baker further discloses that said automatically restoring is performed by a debugger (see, for example, Olsen, column 3, lines 57-67, which shows the debugger) and said

program is absent embedded debug commands (see, for example, Olsen, column 4, lines 12-28, which shows illustrative source code that is absent embedded debug commands).

With respect to claims 53 (currently amended) and 54-58 (previously presented), the claims recite a system that corresponds to the method of claims 45-52 (see the rejection of claims 45-52 above).

With respect to claims 59 (currently amended) and 60-64 (previously presented), the claims recite an article of manufacture that corresponds to the method of claims 45-52 (see the rejection of claims 45-52 above).

Conclusion

7.   Applicant's amendment necessitated any new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707.

The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.
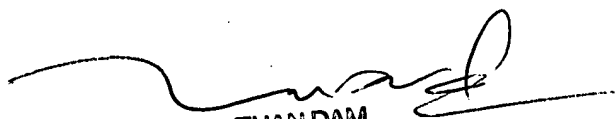
If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private

PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you

would like assistance from a USPTO Customer Service Representative or access to the

automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

                                        MY   :   Michael J. Yigdall
                                                             Examiner
                                                             Art Unit 2192
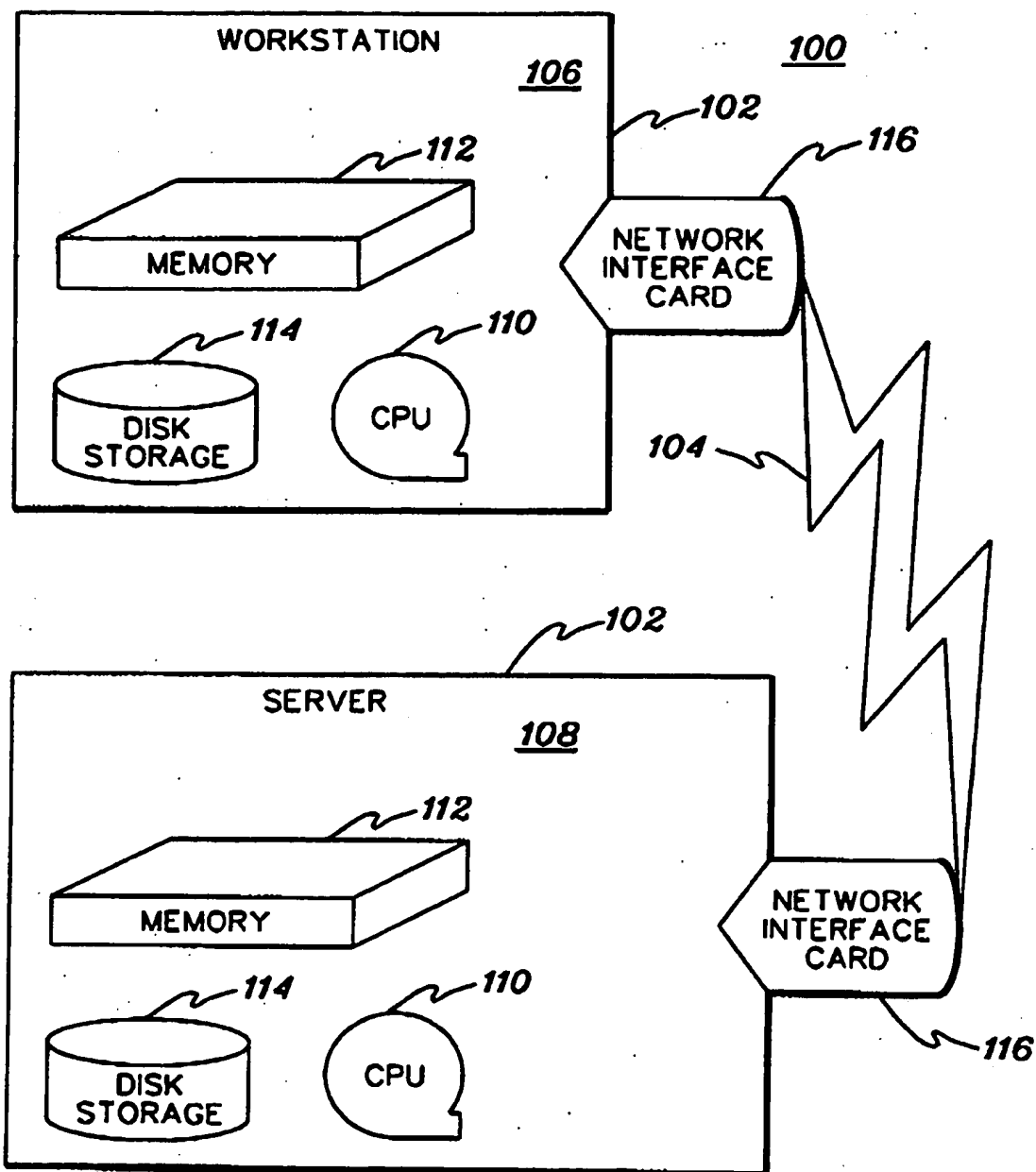
mjy

TUAN DAM
SUPERVISORY PATENT EXAMINER

accepted MY 1/9/07

WORKSTATION
_106_

_100_

_102_

_116_

_112_

MEMORY

_114_

_110_

DISK STORAGE

CPU

NETWORK INTERFACE CARD

_104_

_102_

SERVER
_108_

_112_

MEMORY

_114_

_110_

DISK STORAGE

CPU

NETWORK INTERFACE CARD

_116_

fig. 1A